by Elettronica 2000

N. 27 - MAGGIO 1986 - L. 9.000
Sped. in abb. post. Gr. III/70

LOAD'N'RUN

RIVISTA DI PROGRAMMI SU CASSETTA PER IL TUO COMPUTER

SALVATAGGIO SPAZIALE - LABIPAZZO - ROAD STAR - ASCOT - DJ TELEVISION
RIVENDELL - MOTOR - SPIA - BERNIE - MAGIC LAND - DOMINO - SAVE THE PRINCESS
SCREEN TOOL - SPECTRUM 2 VOCI - SPECTRUM SPRITES - DENUNCIA DEI REDDITI

LOAD'N'RUN

la più bella rivista
di programmi su cassetta
offre

GRATIS
UNA SPLENDIDA MAGLIETTA
a chi si abbona per 6 o per 12 numeri, a prezzo scontato!

PIÙ IN REGALO

SUPERLOAD

una cassetta speciale con tre favolosi programmi: per disegnare, per suonare, per parlare con il tuo Spectrum

ABBONARSI CONVIENE

- dodici numeri con dodici fantastiche cassette, solo L. 90.000. Sei numeri, sei cassette L. 50.000
- risparmi molti soldi e ricevi Load'n'Run direttamente a casa prima che esca in edicola
- hai diritto a ricevere in regalo la maglietta indossata dalla ragazza nella foto e la cassetta SuperLoad con tre eccezionali programmi
- potrai godere della consulenza gratuita della redazione per informazioni tecniche, novità di mercato, recensione di tuoi programmi

È PROPRIO UN'OCCASIONE DA NON PERDERE!

per abbonarsi basta inviare (puoi richiederlo in qualunque ufficio postale) un vaglia postale ordinario compilato come qui a fianco indicato (L. 50.000 per 6 numeri).

Riceverai subito la maglietta e la cassetta SuperLoad e naturalmente mese per mese il tuo sempre più ricco Load'n'Run.

DECIDI SUBITO: IL PREZZO POTRÀ SUBIRE PRESTO UN AUMENTO. PERCHÉ ATTENDERE?! CIAO DA TUTTI NOI DELLA REDAZIONE

WT courtesy

N. 27 - MAGGIO 1986

Direttore
Mario Magrone

Redattore Capo
Sira Rocchi

Direzione Tecnica
Nadia Marini

Stampa
Garzanti Editore S.p.A.
Cernusco S/N (MI)

Distribuzione
SO.DI.P. Angelo Patuzzi spa

Amministrazione, Redazione, Pubblicità: Arcadia s.r.l., C.so Vittorio Emanuele 15, 20122 Milano. Una copia lire 9.000, arretrato lire 10.000. Fotocomposizione: Composit. Selezione colori e fotolito: Eurofotolit. Stampa: Garzanti Editore S.p.A. Milano. Distribuzione: SO.DI.P. Angelo Patuzzi spa, Via Zuretti 25, Milano. Load 'N' Run è un periodico mensile registrato presso il Tribunale di Milano al numero 580 in data 24 dic. 83. Resp. Mario Magrone. Spedizione in abbonamento postale Gr. III/70. Pubblicità inferiore al 70%. Tutti i diritti sono riservati per tutti i paesi. Manoscritti, disegni, fotografie e programmi inviati non si restituiscono anche se non pubblicati. Rights reserved everywhere.

LOAD'N'RUN

RIVISTA SU CASSETTA DI PROGRAMMI PER COMPUTER

SOMMARIO

- SALVATAGGIO SPAZIALE
- LABIPAZZO
- ROAD STAR
- ASCOT
- DJ TELEVISION
- RIVENDELL
- MOTOR
- SPIA
- BERNIE
- MAGIC LAND
- DOMINO
- SAVE THE PRINCESS
- SCREEN TOOL
- SPECTRUM 2 VOCI
- SPECTRUM SPRITES
- DENUNCIA DEI REDDITI

Per caricare i programmi riavvolgete il nastro e date LOAD" ", per uscire dai giochi togliete l'alimentazione e digitate nuovamente LOAD" ". Le cassette utilizzate per registrare i programmi sono tutte di ottima qualità: assicuriamo comunque (rivolgetevi direttamente alla redazione *esclusivamente scrivendo*) la sostituzione di eventuali cassette difettose. Lo ZX Spectrum è un computer prodotto dalla Sinclair Research Ltd. (UK). È distribuito in Italia dalla GBC Italiana SpA.

GAME

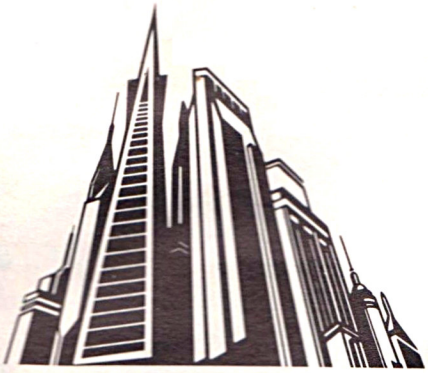
SALVATAGGIO SPAZIALE (48K)

Il temibile Tetrix ha preso come ostaggi alcuni dei tuoi compagni di equipaggio. Tu, in veste d'eroe, hai deciso di attaccare la base di Tetrix per portare in salvo i tuoi amici. Dopo aver abbattuto con il cannone laser le botole dell'astronave, dovrai entrare nella navicella che tiene rinchiusi gli ostaggi avvicinando il corpo dell'eroe alle botole aperte. A questo punto uccidi quanti più robot possibile per poter trarre in salvo il tuo compagno. Se riuscirai nella tua missione passerai ad un nuovo attacco. Avrai bisogno di un po' di fortuna; le operazioni che dovrai compiere non sono facili!

Tasti:
Caps s.,X,V,N,Symbol s.= sinistra
Z,C,B,M,SPACE= destra
da A ad Enter= giù
da Q a P= su
da 1 a 0= fuoco

GAME-ABILITÀ

LABI PAZZO (48K)



Dopo che hai selezionato i tasti con l'opzione 'C', devi riuscire a far cadere i barili nelle nove celle poste in basso al video. Quando i barili cadono attraverso il labirinto, passano per delle porticine che li fanno rimbalzare. Il tuo compito è di studiare il labirinto e provare a determinare la destinazione di ogni barile. Per ogni livello avrai a tua disposizione 20 barili. Se sei molto bravo potrai arrivare fino al settimo livello. Utilizzando i tasti di su e giù potrai cambiare la disposizione del labirinto. Premendo il tasto destra o sinistra sceglierai da quale buca far cadere il barile.

Tasti:
C= permette di definire i tasti
S= start gioco
E= cambia la direzione delle porte (l'attivazione di questo comando fa perdere un barile)

CORSA

ROAD STAR (48K)

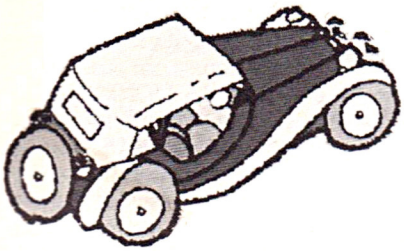
di S. Cacciotti

Un gioco mozzafiato per tutti gli amanti della guida, gli ingredienti per vincere sono: prontezza di riflessi e colpo d'occhio. Stai attento agli ostacoli se non vuoi sfasciare la tua auto e per favore... rispetta i pedoni!

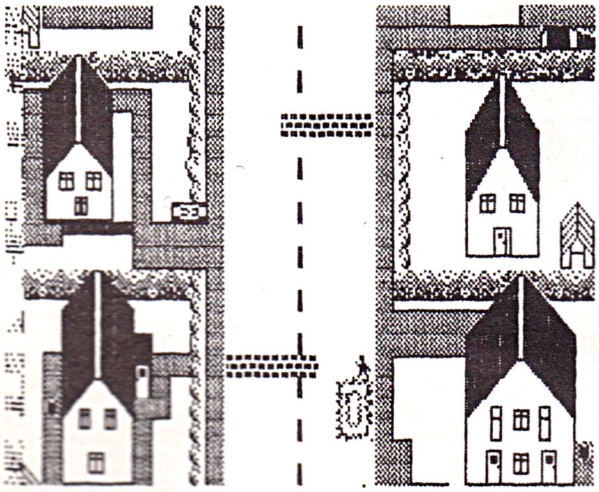
Gli ostacoli restano uguali fino a quando non distruggi l'auto, i percorsi sono di difficoltà crescente e più resisti più punti prendi (fai una sfida con i tuoi amici).

I tasti da usare sono solo due: "L" per sinistra ed "ENTER" per destra, ma stai attento a non rompere la tastiera del tuo computer nella foga di vincere!

Per risparmiare memoria l'autore ha utilizzato diversi artifici di programmazione: ad esempio, invece di scrivere PAPER 0 (colore dello sfondo: nero) ha scritto PAPER NOT PI. Infatti per il com-



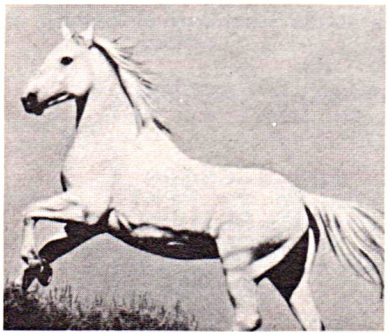
puter la negazione di un numero (in questo caso pi greco) è uguale a 0; quindi, invece che usare i cinque bytes più il numero di bytes corrispondenti alle cifre del numero (in questo caso una cifra), ne usa solo due: uno per NOT e l'altro per PI. Se invece di NOT PI avesse scritto solo BIN avrebbe risparmiato ancora un byte di memoria.



```
   1 PAPER NOT PI: BORDER NOT PI
: INK NOT PI: CLEAR VAL "25000":
 LOAD ""CODE : POKE VAL "23675",
88: POKE VAL "23676",127: LET r=
NOT PI: LET p=NOT PI
   5 CLS : PRINT AT 2,12; INK 2;
 PAPER 5; FLASH 1;"ROADSTAR";AT
5,15; INK NOT PI; PAPER 7;"by";A
T 10,8; PAPER 3;"Mr Byte Softwar
e.";AT 16,12-INT (LEN STR$ r/2);
 INK 6;"RECORD: ";r
   6 PAUSE NOT PI: GO SUB VAL "3
10"
  10 RANDOMIZE USR VAL "32000":
RANDOMIZE USR VAL "25001": GO SU
B VAL "400"
  30 RANDOMIZE USR VAL "32000":
RANDOMIZE USR VAL "25021": GO SU
B VAL "400"
```

ASCOT (48K)

di P. e K. Di Martino



Ascot è un gioco di simulazione ambientato nel mondo delle corse e prende il nome dalla cittadina inglese che possiede uno degli ippodromi più famosi d'Europa.

Insieme con i tuoi amici devi scommettere sui sei cavalli che, di volta in volta, partecipano alla corsa. Possono gareggiare fino a venti giocatori, ciascuno dei quali può fare più di una scommessa per volta. All'inizio bisogna inserire il tempo, in minuti, per cui sarà possibile fare scommesse; ciò dipende naturalmente dal numero delle persone che partecipano al gioco. La fase delle scommesse può comunque essere interrotta rispondendo con "STOP" alla domanda: "Chi scommette?".

Il gioco prosegue con la presentazione del totalizzatore, che dà le quote relative ai sei cavalli che partecipano a quella corsa, e con la presentazione degli ultimi piazzamenti ottenuti dai cavalli in ga-

```
T O T A L I Z Z A T O R E

CAVALLO              VINC.    PIAZZ.

1 )BENHUR            2:1      0.8:1
2 )ZORRO             8:1      3.2:1
3 )BELZEBU'          2:1      0.8:1
4 )CROSTATA          4:1      1.6:1
5 )VIDALIA           2:1      0.8:1
6 )PANIGALE          5:1      2  :1

CAVALLO              MAURO
PANIGALE             SCOMMETTE LIRE
                     10500    SU
ULTIMI PIAZ.         BENHUR
1 2 1   3 3          VINCENTE
```

```
8015>OVER 0: INK 0: PAPER 4: PRI
NT BRIGHT 1;AT 1,1; PAPER 5;"DI
MARTINO"'AT 2,1;"SOFT  TEAM";AT
3,1;" PRESENTA ";AT 19,5; INVERS
E 1; BRIGHT 1;"NEL REGNO DELL'IP
PICA":: PRINT INK 2;AT 9,5;"
SCOT  A
                            "
8020>PAPER 7: KEYWORDS 1
8050 RETURN
8500 POKE 23296,OR: POKE 23297,V
E: POKE 23299,175-8*YP: POKE 232
98,XP*8+1
8510 FOR A=1 TO LEN R$: POKE 233
00,CODE R$(A)-32: LET L=USR 4850
0: IF PEEK 23310 THEN LET L=USR
```

ra. Premendo un tasto si entra nella fase di scommessa: bisogna inserire il nome del giocatore che scommette, ed il computer visualizza il suo capitale (all'inizio è di L. 1.000.000). In seguito si inseriscono l'ammontare della scommessa, il numero del cavallo ed infine V o P per Vincente o Piazzato. Il computer aggiorna il totalizzatore e si rimette in attesa di una nuova scommessa fino a quando il tempo non è scaduto. Il programma prosegue presentando un quadro riassuntivo delle scommesse fatte, ed infine inizia la gara dei sei cavalli.

Al termine della corsa vengono presentate la classifica ufficiale e la situazione finanziaria di ciascun giocatore, con la cifra vinta o persa nella manche.

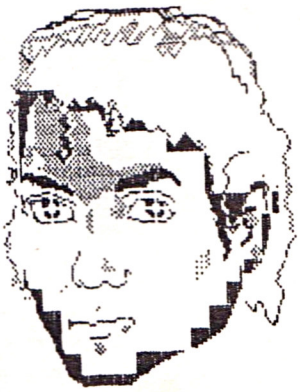
Per scrivere "Ascot" è stato usato il Betabasic. Questo stupendo linguaggio (chiamarlo utility è limitante), contiene dei comandi che elevano all'ottava potenza la versatilità e la facilità d'uso del nostro piccolo Spectrum. I nuovi comandi si ottengono in modo espanso con il cursore in modo "G" (Graphics): al posto dei caratteri grafici vengono stampati i comandi del Superbasic. Ma niente paura! I caratteri grafici ci sono ancora! Per riottenerli basta dare il comando KEYWORDS 0 (tasto "8" con il cursore in "G" + 0). Per ritornare invece al modo comandi bisogna scrivere KEYWORDS 1.

Per vedere il listato dai il BREAK *tenendo premuto* il tasto fino a quando il programma si ferma.

VIDEO CLIP

DJ TELEVISION (48K)

di F. Curridori



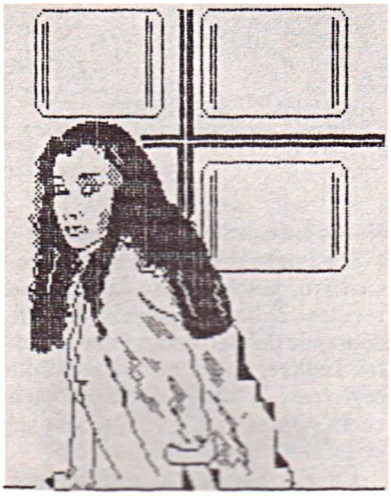
Questa volta il tuo piccolo (ma potente) compagno di tanti giochi ed avventure ha deciso di cambiare: vuole portarti a vedere un concerto dal vivo. Tieniti forte, perché la stella, il big che andrai a vedere è proprio lui, Michael Jackson nel suo stupendo video Triller!

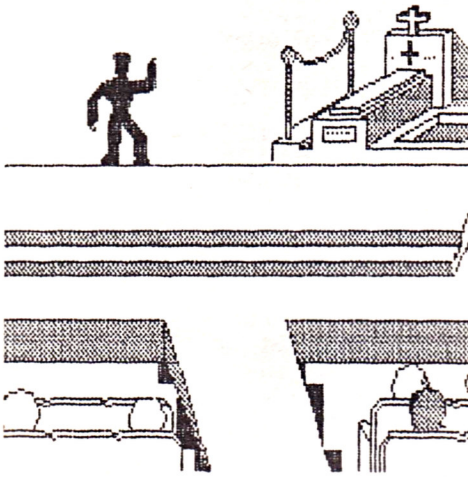
E non è tutto! Appena finito di caricare la prima parte del programma (ricordati di fermare il

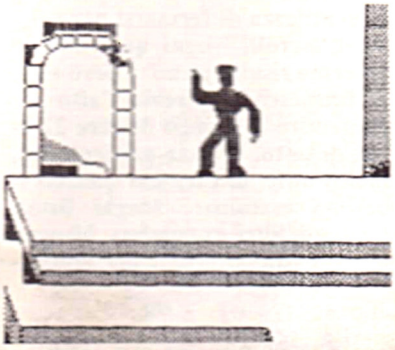
```
4500>FOR s=0 TO 69 STEP .5: BEEP
 .0015,s: OUT 254,s: NEXT s: RET
URN
4559 REM bordo1
4560 RESTORE 4565: FOR u=65400 T
O 65416: READ d: POKE u,d: NEXT
u: RANDOMIZE USR 65400: RETURN
4565 DATA 14,255,6,255,120,211,2
54,16,251,13,32,246,62,1,211,254
,201
4569 REM bordo2
4570 RESTORE 4575: FOR u=65400 T
```

registratore a questo punto), compare la bellissima Kaj Rasch che con voce sintetizzata ti dice: "Salve sono Kaj Rasch e vi presento il nuovo video di Michael Jackson". Il resto? È tutto da vedere!!!

Stai attento all'effetto "discoteca" sul bordo dello schermo! Se vuoi sapere come è stato ottenuto, ferma il programma e guarda la linea 4500: la routine è molto semplice, infatti utilizza un ciclo FOR... NEXT; nel ciclo è presente un BEEP di durata molto breve che dà l'effetto sonoro sintetizzato, ma cos'è che fa cambiare il colore del bordo se non c'è l'istruzione BORDER? Al suo posto viene usato OUT 254, n: così come ci sono 65536 indirizzi di memoria, nello Spectrum si trovano altrettante porte di INPUT/OUTPUT, collegate alle diverse periferiche del computer (tastiera, registratore, video etc.). La porta 254, appunto, in uscita controlla il colore del bordo, l'altoparlante e la presa MIC. "n" deve essere in un range da 0 a 255 che corrisponde a 32 cicli di cambiamento dei colori presenti sulla tastiera da 0 a 7. ■







AVVENTURA

RIVENDELL (48K)

di M. Poli

NewSoft

Questa stupenda avventura è liberamente tratta dal primo dei sei libri che compongono la trilogia del Signore degli Anelli. È in tempo reale (susseguirsi durante la vicenda di giorno e notte), ed è composta da oltre 200 locazioni, la maggior parte delle quali è provvista della carta topografica del luogo, realizzata secondo la simbologia internazionale.

Bilbo, il personaggio che nella famosa avventura "The Hobbit" doveva conquistare il tesoro protetto dal drago, trovò per caso un anello che rendeva invisibile chi lo portava. Ebbene, nella festa del 111° compleanno di Bilbo, Gandalf, stregone amico degli hobbit, strappa l'anello a Bilbo e lo dona al nipote dell'anziano hobbit, Frodo, raccontandogli la storia dell'anello. Esso fu forgiato da Sauron, un oscuro signore del male, assieme ad altri due. Sauron era, dei tre, il più potente. Ora Sauron rivuole il suo gioiello ed ha inviato 9 fidi cavalieri, i cavalieri neri, ad inseguire Frodo ed ucciderlo.

Tu sei, ovviamente, Frodo; con tre amici (Sam, Merry e Pipino) devi raggiungere la città di Gran Burrone (Rivendell) dove troverai gli elfi tuoi amici ed il loro capo,

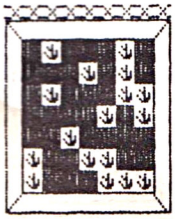
```
9010>IF INKEY$="e" AND ed>VAL "0
" THEN LET xd=ed: POKE 23692,255
: PRINT s$;":"Andiamo a est.":
GO SUB VAL "9019": GO TO xd
9012 IF INKEY$="n" AND nd>VAL "0
" THEN LET xd=nd: POKE 23692,255
: PRINT s$;":"Andiamo a nord.":
 GO SUB VAL "9019": GO TO xd
```

```
9013 IF INKEY$="L" THEN LOAD "d"
 DATA z(): LET xd=z(VAL "1"): LE
T v7=z(VAL "2"): LET v5=z(VAL "3
"): LET pul=z(VAL "4"): LET Mer=
z(VAL "5"): LET Mag=z(VAL "6"):
LET cow=z(VAL "7"): LET oi=z(VAL
 "8"): LET sot=z(VAL "9"): GO SU
B 5000: GO TO xd
```

Elrond.

Il personaggio viene comandato attraverso la pressione di un solo tasto per volta:

e - vai ad est
o - vai ad ovest
s - vai a sud
f - sfila anello
z - copy schermo
C - nasconditi
E - esamina
p - parla/urla
S - si (risposta)
j - salta
Z - salva
n - vai a nord
i - infila anello
a - attacca
d - dormi
L - carica
c - canta



N - no (risposta)

Il numero dei comandi non è restrittivo, poiché molte azioni sono regolate dalla risposta che dai alle domande dei personaggi che incontri. Inoltre, con Z ed L è possibile registrare la situazione e poi caricarla. Questo però solo se non sei invischiato in qualche pericolo.

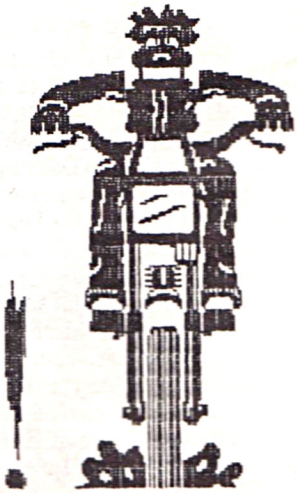
Lo ZX Spectrum, a differenza di tutti gli altri tipi di computer, ha la gentilezza di fermarsi per chiedere "scroll?" ogni qualvolta lo schermo risulta pieno, e deve essere fatto scorrere verso l'alto per consentire la stampa di altre 22 linee di testo. Questa particolarità, molto utile ad esempio quando si devono esaminare lunghi listati (con gli altri computers bisogna avere i riflessi pronti per precipitarsi in tempo sul tasto che ferma il listing), risulta fastidiosa quando i testi da stampare sono brevi ed intervallati da pause come nelle avventure.

L'autore del programma ha abilmente rimediato a questo inconveniente andando a ritoccare la variabile di sistema SCR CT (Contatore di scroll: indirizzo 23692): questa variabile contiene sempre 1 + il numero delle linee che il computer deve far scorrere verso l'alto prima di fermarsi e chiedere "scroll?". Aggiornando costantemente questa variabile con 255 (POKE 23692,255), lo schermo continuerà a scorrere verso l'alto senza mai fermarsi per chiedere "scroll?". ■

MOTOR (48K)

di Raffaele Stefano

Se sei un patito di moto cross o ti piace zigzagare nel traffico cittadino facendo impazzire quei rompiscatole di automobilisti, questo è il programma che fa per te. Segui con attenzione i consigli del tuo Spectrum. L'uso del programma è molto semplice: quando



```
AF97 2A0056    ld hl,(5600)
AF9A 110040    ld de,4000
AF9D 7E        ld a,(hl)
AF9E FE00      cp 00
AFA0 2805      jr z,AFA7
AFA2 12        ld (de),a
AFA3 13        inc de
AFA4 23        inc hl
AFA5 18F6      jr AF9D
AFA7 23        inc hl
AFA8 7E        ld a,(hl)
AFA9 FE00      cp 00
AFAB 280A      jr z,AFB7
AFAD 47        ld b,a
AFAE 3E00      ld a,00
AFB0 12        ld (de),a
AFB1 13        inc de
AFB2 10FC      djnz AFB0
AFB4 23        inc hl
```

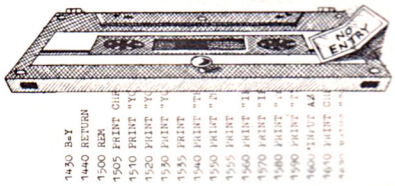
vuoi cambiare pagina premi un tasto. Per trasferire le immagini sul video è stato usato un programma in linguaggio macchina presente in memoria a partire dalla locazione 44951 (AF97 esadecimale). Nel programma si trova un ciclo a contatore (simile al ciclo FOR... NEXT del basic) che utilizza il registro B dello Z80 con l'istruzione

DJNZ.

DJNZ, tradotto, significa: decrementa e salta se non è zero e si riferisce appunto al registro B, che deve essere inizializzato all'inizio del ciclo. L'indirizzo a cui saltare deve essere invece specificato come argomento di DJNZ. Un ciclo del genere in linguaggio macchina ha una velocità di esecuzione elevatissima, ma è limitato dalla capienza del registro B che, essendo ad 8 bits, può contenere 255 come numero massimo.

SPIA (48K)

di Adriano Valle




```
 550>LET a=INT (f/256): LET c=f-
a*256: POKE 23675,c: POKE 23676,
a: PRINT AT 0,0;"ABCDEFGHIJKLMNO
PQRSTU": PRINT f;"      "
 560 PAUSE 0
 565 LET b$=INKEY$
 570 IF b$="e" THEN GO TO 10
 580 IF b$>"0" AND b$<"9" THEN L
ET g=f: LET d=VAL b$: LET w=6553
5: GO TO 520
 590 IF INKEY$="9" THEN LET g=f:
 LET d=-d: LET w=0: GO TO 520
 595 IF INKEY$="0" THEN LET g=f:
 LET d=128: LET w=65535: GO TO 5
20
 600 IF b$="m" THEN GO SUB 620:
GO TO 560
 610 NEXT f
 614 PRINT #1;"Fine!": FOR p=0 T
O 150: NEXT p: GO TO 10
 620 LET e=10: FOR w=175 TO 168
STEP -1: FOR q=0 TO 7: IF POINT
(q,w)=1 THEN PRINT AT e,q;"": G
O TO 640
 630 PRINT AT e,q; PAPER 4;"X"
 640 NEXT q: LET e=e+1: NEXT w
 650 RETURN
```

Ecco una comoda e semplice utility che ti aiuterà a personalizzare i tuoi programmi. Essa è composta di cinque parti che esaminiamo subito una per una. Terminato il caricamento, ecco il menù principale con le opzioni da scegliere:

1) Cerca e cambia le scritte: serve a cercare le scritte ed i caratteri in memoria per poterli poi cambiare. Al primo INPUT "Start" devi inserire l'indirizzo dal quale hai intenzione di partire. Cominceranno a scorrere tre colonne di dati: la prima contiene l'indirizzo, la seconda il valore di PEEK della locazione, la terza il carattere corrispondente al valore della colonna due. Se il valore è minore di 32, nella terza colonna ci sarà il segno — (meno). Premendo P avrai la pausa, che potrai interrompere usando lo stesso tasto. Premendo 1 verrà visualizzato un INPUT al quale dovrai rispondere con il solito indirizzo di partenza ma, questa volta, della frase che vuoi cambiare. Di seguito dovrai scrivere la frase che verrà inserita automaticamente. Il tasto E premuto durante lo scorrimento serve ad uscire per tornare al menù.

2) Cerc disegni: serve appunto a ricercare disegni o parti di disegni residenti in memoria. Dopo aver risposto all'INPUT, sul video appariranno i bytes contenuti in memoria a partire dall'indirizzo dato. Premendo 1 il bordo diventerà verde: significa che sei in modo scroll 32, quindi muovi la figura con step 32. Il tasto 1 fa ritornare lo step ad 1 nel modo scroll 1. Premendo I verrà visualizzata la locazione dalla quale partono i bytes visualizzati in memoria. Premi E per uscire. Usa uno degli altri tasti per far scorrere la figura.

3) Cerca UDG con ingranditore: visualizza gli UDG definiti in memoria ed ingrandisce il primo della fila. I tasti da 1 ad 8 servono per decidere lo STEP, che viene visualizzato sullo schermo. Il tasto M serve per ingrandire il primo carattere che viene segnato sulla linea. Premi E per uscire. Premi uno degli altri tasti per muovere gli UDG con lo STEP definito.

4) Cerca attributi: per ricercare in memoria attributi di disegni o di giochi. I tasti sono gli stessi dell'opzione 2.

5) Leggi header e conta bytes: per misurare la lunghezza dei blocchi e dare le informazioni dell'header. Basta far partire il registratore ed il blocco viene caricato a partire dall'indirizzo 32768. Se si tratta solo di un blocco di DATA, vengo-

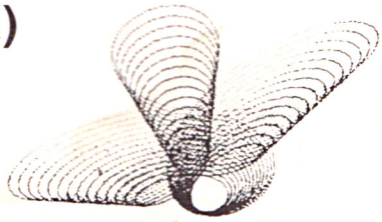
no forniti la lunghezza ed il valore del registro A (per chi se ne intende un po' di salvataggi e caricamenti in linguaggio macchina). Se il blocco è un header, ne vengono letti i dati. Questa funzione è molto utile soprattutto per esaminare quei programmi basic che sono protetti da break e da merge: basta caricarli e guardarseli a partire dall'indirizzo 32768.

Questo è tutto. In caso di errore, dare RUN. Il programma è stato intenzionalmente lasciato aperto per l'utilizzo personale delle singole parti. Se hai il microdrive, puoi breakare al menù principale e salvare il programma su cartridge con: SAVE * "m";1;"SPIA" LINE 0.

Un ultimo consiglio: se devi lavorare su blocchi lunghi che quindi non entrano completamente in memoria, è meglio che tu divida il tutto in due o più parti.

Nonostante la sua potenza, questo programma è interamente scritto in basic. Per poter essere efficiente deve andare però a curiosare nelle aree più nascoste della memoria dello Spectrum. Per cercare ad esempio gli UDG (caratteri grafici definiti dall'utente), deve continuamente cambiare il loro indirizzo per scandagliare le diverse aree della memoria: questo diventa possibile andando a scrivere nella variabile di sistema UDG (variabile a due byte che si trova all'indirizzo 23675) il nuovo indirizzo di partenza dei caratteri grafici. Proprio come si fà con la variabile a due byte CHARS (indir. 23606), quando si vuole usare un set di caratteri definito in RAM.

BERNIE (48K)



Aiuta il povero Bernie ad uscire dai ben 63 schemi che compongono questo divertentissimo gioco.

In ogni schema è necessario impossessarsi di tutti gli oggetti evitando il contatto con gli strani personaggi che vi dimorano e preservando il più possibile la zona colorata, il tutto in un tempo non troppo lungo, ovviamente.

A queste difficoltà si aggiungono, talvolta, dei quadrati magici che impongono una certa direzione al povero Bernie. In alcuni casi può essere utile servirsi dei misteriosi EDIT rotanti che, sebbene pericolosi quando ti cadono addosso, possono bloccare i tuoi inseguitori se usati con astuzia. Inoltre, a differenza di molti altri giochi simili, si può continuare dall'ultimo schema giocato o salvare su cassetta lo schema al quale si è giunti per poter riprendere il gioco successivamente.

Ecco qui di seguito come operare.

Tasti:
S= per iniziare
R= per iniziare da capo
CAPS+L= per caricare uno schema
CAPS+P= per salvare lo schema a cui si è giunti
P - O= destra - sinistra
Q - A= su - giù
H= stop
S= riparte

THE MAGICLAND OF LANDLORDS (48K)

di C. Altieri

Il terribile Landlord Devil si è impadronito del castello del destino dal quale tiranneggia su tutta Magicland tramite i suoi mostruosi servitori. Il Mago Wiz con i suoi poteri cercherà di riportare la pace.

Diventi immortale prendendo il fungo, la fragola, l'amuleto ed il pipistrello bianco.

Le quattro chiavi ti permetteranno di entrare nel castello. Per

sconfiggere il Landlord Devil dovrai aver raccolto energia Omega a sufficienza (l'indicatore lampeggerà).

Tutti gli altri oggetti ti riserveranno gradite sorprese, ma ricordati che devi completare l'impresa prima che finisca il tempo scandito dalla candela posta a destra.

Tasti:
1= Pausa
4= Riparti
X= Destra
Z= Sinistra
SPACE= Salto

Dal listato prodotto dalla stampante non si nota, ma il programmino in basic che carica il gioco in linguaggio macchina è tutto colorato; questo effetto si può facilmente ottenere se, scrivendo un programma, con il cursore in "E" si preme un tasto corrispondente ad un colore (numeri da 1 a 7 e lo 0): il testo scritto successivamente assumerà come colore della carta (PAPER) quello corrispondente al numero premuto. Se invece contemporaneamente al numero si preme CAPS shift, cambierà il colore dell'inchiostro (INK). I numeri 8 e 9, sempre con il cursore in "E", controllano la funzione BRIGHT (8=BRIGHT 1; 9=BRIGHT 0); premuti invece contemporaneamente a CAPS shift corrispondono a FLASH 1 (9) e FLASH 0 (8).

```
  10 INK 7: PAPER 0: BORDER 0: C
LEAR 25000
  20 INK 0: PAPER 0: BORDER 0: C
LEAR 25000
  30 PRINT AT 8,9;"
  ";AT 9,9;          " THE MAGICLAN
D ";AT 10,9;         "
  ";AT 11,9;         " OF  LANDLORD
S ";AT 12,9;         "
  "
  35 PRINT #1;"          © 1985  by
 PROMETEO"
  40 PRINT AT 0,0;: LOAD ""CODE
27904: LOAD ""CODE : POKE 48050,
9: POKE 55601,32: RANDOMIZE USR
27904
9000 LOAD ""CODE 27904: LOAD ""C
```

CLASSIC

DOMINO (48K)

di G. Di Paolo

Giuseppe Di Paolo

FERRARA

Non sarà più necessario usare tante tavolette numerate e lunghi tavoli per giocare a Domino, ora che anche questo gioco gira sul tuo Spectrum!

Le regole del Domino sono molto semplici: devi riuscire a disporre tutte le tue tavolette sul tavolo da gioco unendole, per corrispondenza numerica (es: il 5 con il 5, il 3 con il 3 etc.), a quelle già giocate da te o dal computer.

Tasti:
5= movimento cursore a sinistra
8= movimento cursore a destra
0= conferma la scelta della tavoletta
Q= passa il turno di gioco

Visto come è stata composta la musichetta? Dalla linea 5060 alla 5150 ci sono delle strane variabili stringa: la linea 5150 contiene la variabile con i codici relativi alla nostra breve sigla: il ciclo alla linea 5160 legge i caratteri della variabile e li trasforma in numeri (codici ASCII del carattere corrispondente) utili per far suonare lo Spectrum con la funzione BEEP.

Le altre strane variabili stringa invece contengono i dati necessari per far disegnare allo Spectrum la scritta DOMINO con una analoga routine.

```
5060 LET b$="R>> W>>M*M9>": RETU
RN
5070 LET b$="W9>*M>>F/J": RETURN

5080 LET b$="u>> W>>\%>": RETURN

5090 LET b$="z9>*M>>R/>": RETURN
5100 LET b$="I>> C>>WF/EM>%C>>\9
>7/6M9>": RETURN
5110 LET b$="SQR >> C>>\9>": RET
URN
5120 LET b$=" LINE >> C>>WM%C>>\
9>>%/W9>": RETURN
5130 LET b$=" GO SUB >> W>>\%>":
 RETURN
5140 LET b$=" PAUSE 9>*M>>R/>":
RETURN
5150 LET b$="aaaacfechhhhhjefccc
ccfecamljhfecaaaacfechhhhhjefccc
```

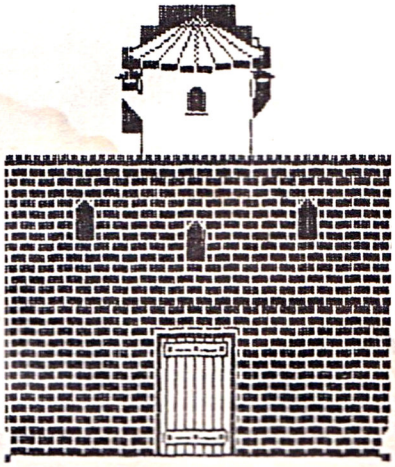
AVVENTURA

SAVE THE PRINCESS (48K)

di M. Salis

```
  69 GO TO 89
  70 CLEAR 46999: LOAD ""CODE 47
000: GO TO 1
  80 SAVE " SAVE THE" LINE 70: S
AVE "PRINCESS"CODE 47000,65530-4
7000
  89 POKE 65084,0: POKE 65085,0:
 POKE 65086,0: POKE 65083,15: PO
KE 65097,1: POKE 65098,1
  90 PAPER VAL "1": CLS : PAPER
VAL "6": INK VAL "1"
  91 POKE 65087,255: RANDOMIZE U
SR VAL "54820": RANDOMIZE USR 54
800
  92 BORDER 0: PAPER 0: INK 7: R
```

La bellissima principessa del castello è vittima in un crudele incantesimo, tu soltanto puoi salvarla. Per farlo devi trovare la



pozione che ha preparato il mago Merlino e uccidere il malefico Antor, quindi baciare la principessa. Nell'avventura avrai bisogno dell'aiuto di Ivan che dovrai cercare fin dall'inizio.

L'azione si svolge in un castello di 141 stanze rappresentate graficamente in alto a sinistra. In alto a destra si trovano gli oggetti raccolti, mentre la parte bassa dello schermo è riservata ai messaggi ed ai comandi, rappresentata da una bellissima pergamena gialla. I comandi sono quelli tipici delle avventure; se ti trovi in difficoltà o non sai cosa fare, puoi sempre chiedere aiuto.

Per comunicare con Ivan devi scrivere: "Ivan: frase"; ad esempio: IVAN: SEGUIMI.

Il programma, scritto quasi interamente in basic, è stato protetto, infatti viene caricato come bytes. Il nome del blocco in questione è "PRINCESS". Se vuoi analizzare il listato scrivi il seguente programmino e ricaricalo:

10 CLEAR 33295 : LOAD "PRINCESS"CODE 33296 : POKE 23635, 23755/256 * INT (23755/256) : POKE 23636, INT (23755/256) : LIST

Attenzione però perché così puoi solo vedere il listato: non puoi modificarlo né farlo girare.

UTILITY

SCREEN TOOL (48K)

di A. Loschiavo

Il programma, utilizzabile anche a mezzo drive, è una multiutility per la gestione dello schermo: spostamenti a mezzo di 4 fantastici scroll (compresi attributi e wrap around), trasferimenti, creazione di finestre, inversione, ingrandimento, stampa su 24 linee ed altre manipolazioni ancora. Un menù iniziale evidenzia l'elenco dei tasti usati e le funzioni relative.

Tasti:

C= per i possessori di drive: fornisce l'elenco dei programmi contenuti nel disco o nella cartuccia in

quel momento utilizzata.
E= richiama lo screen in quel momento locato all'indirizzo 32768.
F= provoca il "mirror" o rovesciamento speculare dell'immagine e la successiva memorizzazione alla locazione 40000 e seguenti.
G= fa comparire, in INK 8 o OVER 1, una griglia rettangolare che delimita la zona di schermo — selezionata a mezzo degli scroll — da ingrandire con l'apposita routine (la griglia si cancella premendo due volte il tasto).
H= memorizzata a partire dalla locazione 40000 l'immagine presente in quel momento.
I= esegue l'inverse video dell'intera schermata (24 linee).
j/J= caricamento di uno screen da nastro (j) o da drive (J).
K= lista del programma con inizio richiesto da Input.
L= memorizza l'immagine a partire dalla locazione 32768.
M= ingrandisce il rettangolo centrale dello schermo creando istantaneamente un nuovo screen a pieno schermo.
Q= fornisce l'elenco delle routine in L/M ed il loro indirizzo di partenza.
s/S= salva su nastro (s) o su drive (S) l'immagine presente sullo schermo.
R= ritorno al menu.
T= richiamo screen caricato inizialmente e su cui si desidera eseguire le elaborazioni. Tale immagine resta permanentemente in RAM fino a quando non viene caricato un altro file sostitutivo (locazioni di riserva 50000/56912).
U= richiamo finestra creata precedentemente con l'utilizzo del tasto W.

```
 541>GO SUB 241: PRINT AT 19,3;
FLASH 1; BRIGHT 1;" SAVING "; F
LASH BIN ;" "; INK 2;k$;" "; I
NK BIN ; PAPER 6;" CODE ";(32768
 AND NOT f1)+(57800 AND f1);",";
len;" "
 561 IF j$="s" THEN SAVE k$CODE
(32768 AND NOT f1)+(57800 AND f1
),len
 581 IF j$="S" THEN SAVE *1;k$CO
DE (32768 AND NOT f1)+(57800 AND
 f1),len
 599 GO SUB 181: GO TO 1000
```

W= creazione di una finestra: posizione e dimensione con i dati forniti in input (coordinate x e y, altezza e larghezza).
Y= richiamo dello screen memorizzato a partire dalla locazione 40000.
Z= esegue l'hardcopy totale della immagine su 24 linee.
1= scroll a sinistra.
2= scroll a destra.
9= scroll verso il basso.
0= scroll verso l'alto.

Per poter essere utilizzato con i microdrives della Sinclair il programma necessita di qualche modifica: infatti le istruzioni usate per far funzionare i vari tipi di Floppy Disk non corrispondono pienamente, nella sintassi, a quelle necessarie per il controllo degli ZX Microdrives. La linea 581, ad esempio, va cambiata in:
581 IF j$="s" THEN SAVE * "m";1;k$ CODE (32768 AND NOT f1)+(57800 AND f1),len

Questa sintassi dei comandi così complicata, in effetti è necessaria all'interface 1 poiché, oltre che i microdrive, deve gestire anche la porta seriale RS 232 e la rete locale che possiedono un set di comandi molto simile.

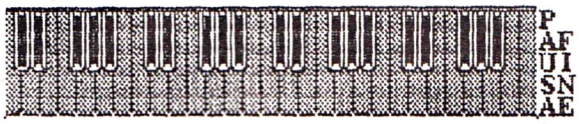
SPECTRUM DUE VOCI (48K)

di F. Rizzo

Ecco un programma stupendo per poterti improvvisare musicista e compositore: per poterlo usare bastano solo cinque tasti: i cursori (5, 6, 7, 8) e lo 0. Lo schermo principale è composto

```
I   VOCE ATTIVATA          SUONA
II  VOCE DISATTIVATA       CANCELLA
COMPONI      TEMPO         CARICA
CORREGGI     TONO          SALVA

MI3MI3MI3MI3MI3MI3MI3- - - - - - - - - -
II  VOCE
SI2RE2MI2MI2MI2MI2MI2- - - - - - - - - -
I   VOCE
          BY FABIO RIZZO
```




```
  30>PRINT AT X,Y; OVER 1; INK 5
;"  ": PAUSE 0: PRINT AT X
;Y; INK 7; OVER 1;"  "

  40 IF INKEY$="6" AND X<7 THEN
LET X=X+2
  42 IF INKEY$="7" AND X>1 THEN
LET X=X-2
```

nella parte alta dal menù con le diverse opzioni, mentre nella parte bassa si trova una rappresentazione grafica della tastiera di un pianoforte. Per selezionare le diverse opzioni devi usare i cursori e premere lo 0 quando la funzione che vuoi usare diventa di colore azzurro. Se vuoi ascoltare subito la melodia contenuta nel programma devi selezionare la prima e la seconda VOCE ed attivarle con il tasto 0, prima di scegliere l'opzione SUONA.

PER COMPORRE UNA MELODIA

Quando si inizia a comporre, bisogna chiedere sempre per prima l'opzione COMPONI che pone a 0 tutte le variabili e posiziona il cursore sulla tastiera musicale. Naturalmente puoi comporre la musica di una sola voce alla volta, quindi solo una delle voci deve essere attivata, altrimenti il computer ti rimanda al menù principale. Se vuoi tornare al menù devi spostare il cursore sulla parola FINE scritta in verticale e premere come sempre lo 0.

Le altre possibilità offerte sono:
CANCELLA: quando viene selezionata bisogna attendere che la scritta ridiventi azzurra, poi passare all'opzione COMPONI: le note ora composte si sovrappongono alla melodia già composta.
CORREGGI: compare un quadrato sulla nota selezionata; per spostarlo usa i cursori poi premi zero e sostituisci la nota sbagliata con un'altra. Questo cursore è importante perché indica fino a dove il computer deve suonare la melodia.
TEMPO: definisce il tempo della melodia con valori da 0 a 255; si cambia con i cursori 6 e 7.
TONO: definisce la tonalità e può essere da 0 a 20.
CARICA: carica una melodia precedentemente composta.
SALVA: registra la melodia composta.

A proposito del programma: come fà il nostro amico a far diventare azzurre le scritte senza prima cancellarle?

Semplice, gli sovrascrive, con OVER 1, otto spazi di colore azzurro! Dà un'occhiata alla linea 30 del listato.

SPECTRUM SPRITES

(48K)

di K. ed R. Di Martino

Il programma consiste essenzialmente in una routine in linguaggio macchina che permette di usare sprites di numero e dimensioni qualsiasi, di guidarne il movimento sullo schermo e di rilevare eventuali collisioni, il tutto molto semplicemente, usando cioè il Basic.

I dati di cui si serve la routine in linguaggio macchina sono infatti contenuti in tre matrici alfanumeriche (A$, B$ e C$), nella forma che sarà in seguito specificata.

È stato inoltre realizzato un programma prevalentemente in Basic per facilitare il compito della realizzazione e della programmazione degli sprites.

Le tre matrici A$, B$ e C$ devono essere, nell'ordine, le prime tre variabili definite dal programma basic.

Descriviamo ora la funzione delle tre matrici A$, B$ e C$.

La prima ha due dimensioni: la prima corrisponde al numero di

sprites che si usano, la seconda è fissa: 35. Questa matrice contiene le informazioni da e per la routine di movimento degli sprites.

La matrice B$ contiene le informazioni per la gestione della tastiera. Ha due dimensioni: la prima corrisponde al numero di tasti definiti, la seconda è fissa: 5.

La matrice C$ contiene le informazioni circa la logica di gioco. Ha anch'essa due dimensioni: la prima che corrisponde al numero di confronti che servono, la seconda fissa: 5.

Vediamo ora il funzionamento della routine in linguaggio macchina.

Essa gestisce le seguenti funzioni degli sprites:
- Abilitazione
- Posizione
- Movimento
- Collisioni

Inoltre gestisce una routine di scansione della tastiera ed un'altra per la logica di gioco.

ABILITAZIONE

Uno sprite è abilitato, e quindi visibile, se il 1° elemento della variabile che lo definisce (quindi A$(n,1)) è uguale a CHR$ 1.

Se A$(n,1)=CHR$ 0 non è abilitato. Se durante lo svolgersi del programma si ha la necessità di disabilitare uno sprite, si può modificare questo elemento; se lo si rende =CHR$ 0, lo sprite rimane sullo schermo anche se è disabilitato; per cancellarlo, fare A$(n,1)=CHR$ 2.

POSIZIONE

Lo sprite può essere posizionato fornendo le coordinate in alta risoluzione (asse delle x:da 0 a 255 andando da sinistra verso destra Asse delle y:da 0 a 191 andando dall'alto in basso).

MOVIMENTO

Gli sprites possono avere due tipi di movimento: continuo, oppure guidato dai tasti.

La velocità di movimento è regolata dalla distanza in pixel fra due successive apparizioni dello sprite. Il movimento verso sinistra e verso l'alto implica uno spostamento negativo sull'asse delle x e delle y, e perciò si ottiene, rispettivamente, con A$(n,9)=CHR$ (256-velocità desiderata) e A$(n,10)CHR$ (256-velocità desiderata).

Si possono stabilire i limiti dello spostamento di ogni singolo sprite, inserendo i valori massimi e minimi delle coordinate che non possono essere superati. Se lo sprite raggiunge questi limiti si ferma, e l'elemento 6 della variabile che lo rappresenta assume valore diverso da 0 (A$(n,6))=CHR$ 1 significa raggiunto x max; =CHR$ 2 significa raggiunto x min; =CHR$ 3 significa raggiunto y max; =CHR$ 4 significa raggiunto y minimo.

COLLISIONI

La collisione viene riconosciuta testando gli attributi della posizione su cui si muove lo sprite. È possibile considerare isolatamente ink paper, bright e flash, oppure loro combinazioni o tutti insieme, utilizzando la cosiddetta maschera degli attributi, (cioè l'elemento 19 (A$(n,19):bit 7 on=flash testato, bit 7 off=flash non considerato; bit 6 per bright; bit 5-4-3 per paper; bit 2-1-0 per ink). Il valore da confrontare è contenuto in A$(n,35).

Il test è positivo, cioè viene riconosciuta la collisione, a scelta, se il valore testato è:

1) maggiore
2) minore
3) uguale
4) diverso

} da quello di A$(n,35)

La scelta fra questi tipi di test si fa tramite l'elemento A$(n,34).

Esempi: se A$(2,19)=CHR$ 7;A$(2,35)=CHR$ 2;A$(2,34)=CHR$ 40 significa: se l'inchiostro della posizione su cui si muove lo sprite n.2 è rosso, allora c'è collisione; oppure: A$(3,19)=CHR$ 128;A$(3,35)=CHR$ 1;A$(3,34)=CHR$32 significa: se flash della posizione su cui si muove lo sprite n.3 è diverso da 1 allora c'è collisione. E così via.

In caso di collisione l'elemento A$(n,20) contiene il valore dell'attributo della posizione che ha provocato la collisione.

CONSEGUENZE DELLA COLLISIONE

Se c'è stata una collisione, si possono scegliere diversi tipi di conseguenza:

1) Lo sprite continua a muoversi, ma viene settato l'elemento A$(n,20) (vedi sopra).
2) Lo sprite si ferma.
3) Sparisce.
4) Si trasforma (in questo caso viene utilizzato l'elemento A$(n,33) che rappresenta il 1' carattere dello sprite dopo la trasformazione)
5) Rimbalza

Si può aggiungere un effetto sonoro sommando 8 ai numeri precedenti. La scelta tra queste diverse conseguenze della collisione si fa con l'elemento A$(n,23).

GESTIONE TASTIERA

Si può definire per ciascun tasto una funzione; la routine che riconosce la pressione dei tasti può modificare qualunque elemento della matrice A$ (per esempio, per ottenere il movimento orizzontale, l'elemento 9 della variabile dello sprite controllato dal tasto può assumere il valore corrispondente alla velocità di spostamento desiderata). Una opzione di questa routine permette di effettuare dei cambiamenti relativi della variabile scelta, cioè di aggiungere o sottrarre un valore definito da quello che vi era in precedenza contenuto.

La routine può, inoltre, cedere il controllo a una linea basic nel caso si desiderino altri tipi di reazione alla pressione di un tasto, o attivare lo sparo. Come detto prima, per usare questa routine i dati devono essere contenuti nella matrice B$.

Esempi:B$(1,1)=CHR$ 2;B$(1,2)=CHR$ 2; B$(1,3)=CHR$ 3; B$(1,4)=CHR$ 5; B$(1,5)=CHR$ 7 equivale al basic: IF INKEY$ "S" THEN LET A$(3,5)=CHR$ 7.

Oppure: B$(2)=CHR$ 2+CHR$ 2+CHR$ 3+CHR$ 133+CHR$ 7 equivale al basic: IF INKEY$ ="S" THEN LET A$(3,5)=CHR$ (CODE A$ (3,5) +7).

Oppure: B$(2)=CHR$ 2+CHR$ 2+CHR$ 0+CHR$ 20+CHR$ 10 significa in basic: IF INKEY$="S" THEN GOTO2580 (256*10+20).

Oppure: B$(3)=CHR$ 2+CHR$ 2+CHR$ 131+CHR$ 2+CHR$ 2 significa: IF INKEY$="S" then attiva sprite per

sparo e inizializzalo con x=CODE A$(3,9)+2 e y CODE A$ (3,10)+2.

LOGICA DI GIOCO

Questa parte della routine fa dei confronti sul valore delle variabili della matrice A$, con una struttura del tipo IF... THEN.

Tipi di controllo:

1) IF un elemento della matrice A$=un dato numero THEN...

2) IF un elemento della matrice A$ è diverso da un dato numero THEN...

Tipi di conseguenza:

1) Sostituzione di un elemento della variabile che definisce lo sprite testato con un nuovo valore;

2) Modifica relativa di un elemento della variabile che definisce lo sprite testato, con aggiunta o sottrazione di un dato valore a quello che vi era in precedenza contenuto;

3) Salto a numero di linea del programma basic (il n. di linea deve essere inferiore a 256).

I dati per l'uso di questa routine sono contenuti nella matrice C$.

Esempi: C$(1,1)=CHR$ 2;C$(1,2)=CHR$ 3;C$(1,3)=CHR$ 4;C$(1,4)=CHR$ 0; C$(1,5)=CHR$ 5 significa:IF A$(2,3)=CHR$ 4 THEN GOTO 5.

Oppure:C$(2)=CHR$ 3+CHR$ 135+CHR$ 15+CHR$ 5+CHR$ 2 SIGNIFICA:IF A$(3,7)=CHR$ 15 THEN LET A$(3,5)=CHR$ (CODE A$(3,5)+2).

Oppure: C$(1)=CHR$ 3+CHR$ 68+CHR$ 12+CHR$ 0+CHR$ 255 significa:IF A$(3,4) è diverso da CHR$ 12 THEN GOTO 255.

SALTO A NUMERO DI LINEA BASIC

Per attivare questa funzione si usa la routine che inizi. a 51099: questa restituisce il numero della linea basic in cui si trova la chiamata con USR, oppure il numero di linea a cui si deve saltare per il verificarsi di una delle condizioni prima descritte (un esempio d'uso: GOTO USR 51099).

ALTRE INFORMAZIONI

Lo schermo su cui si svolge il gioco deve essere già presente nella memoria video quando si fa partire la routine in linguaggio macchina. In seguito, si potrà stampare qualsiasi cosa sullo schermo, ma per modificare gli attributi bisogna usare il seguente sistema: Inserire nel programma basic la linea seguente: DEF FN A(a,b,c,d,e)= USR 49800 e in seguito: RANDOMIZE FN A(x,y,l,a,attr) dove x e y sono le coordinate in alto a sinistra dell'area di schermo da modificare; 1 ed a sono la larghezza e l'altezza di questa area; attr è il nuovo valore dell'attributo.

Per regolare la velocità generale della routine di movimento: poke nella locazione 49917 valori da 1(veloce) in su.

Per regolare repeat tasto di fuoco: poke 49922 da 1 (veloce) in su.

Per regolare velocità dei proiettili: poke 49921 da 1 (lento) in su.

Per attivare la routine: RANDOMIZE USR 49924.

Per disattivarla: RANDOMIZE USR 50005.

PROGRAMMA BASIC

Il programma basic presenta alcune opzioni utili per l'uso della routine in linguaggio macchina.

Il menù è il seguente:

1) sprite editor
2) logica di gioco
3) definizione tasti
4) disegno sprite
5) prova
6) fine
7) save-load caratteri
8) save-load situazione
9) istruzioni

SPRITE EDITOR

Lo sprite editor permette di creare la matrice A$ con tutte le informazioni necessarie. L'opzione chiede prima di tutto se si vuol continuare un lavoro già iniziato o se si ricomincia; in seguito chiede se si vuol definire uno sprite dell'area sparo; in caso affermativo passa a chiedere le informazioni necessarie a programmare uno sprite di sparo, altrimenti chiede se si vuol programmare uno sprite normale. Al termine viene presentato un quadro riassuntivo del lavoro svolto con la richiesta di eventuali modifiche.

LOGICA DI GIOCO

Questa opzione serve a preparare la matrice C$. Anche qui si chiede se si vuol riprendere un lavoro già iniziato o se si ricomincia da capo. In seguito vengono richieste le altre informazioni necessarie alla corretta preparazione della matrice C$.

DEFINIZIONE TASTI

Questa opzione serve a preparare la matrice B$. Si inizia come per le prime, scegliendo se continuare o ricominciare. In seguito viene chiesta la posizione del tasto da definire (riga sulla tastiera, posizione all'interno della riga). Si deve far riferimento alla tastiera dello SPECTRUM normale (40 tasti).

DISEGNO SPRITE

Dà la possibilità di creare gli sprite nel modo in cui saranno utilizzati dalla routine in linguaggio macchina. Permette di realizzare sprites fino a 4x3 caratteri e li visualizza ingranditi 8 volte durante la fase di creazione. I tasti attivi in questa funzione vengono visualizzati premendo "H". Lo schema tipico di costruzione di uno sprite, per esempio di 3x2 caratteri, è il seguente:

A C E
B D F

PROVA

Questa opzione permette di provare quanto si è fatto, attivando la routine in linguaggio macchina dopo aver risistemato le variabili in memoria. Per terminare premere "X".

FINE

A lavoro concluso, salva su nastro le matrici A$, B$ e C$, oltre al nuovo set di caratteri creato. In seguito, queste matrici potranno essere caricate dal programma utilizzatore con LOAD "" DATA A$(): LOAD "" DATA B$(): LOAD "" DATA C$(): LOAD "" CODE.

SAVE-LOAD CARATTERI

Permette il save o il load dei soli caratteri grafici.

SAVE-LOAD SITUAZIONE

Permette di interrompere il lavoro per riprenderlo in un secondo momento.

Chiariamo infine cosa si intende

per "area sparo", e per "sprite per sparo".

Gli sprites usati dalla routine possono essere suddivisi in due categorie: sprites normali e sprites per sparo. Questi ultimi rappresentano i "proiettili" che vengono lanciati in caso di sparo. Sono attivati quando viene premuto il tasto di fuoco.

Le coordinate di partenza corrispondono a quelle dello sprite che spara + i valori contenuti nella variabile della matrice B$ che definisce il tasto di sparo.

La velocità dello sprite di sparo è uguale a quella dello sprite di partenza moltiplicato per il valore contenuto nella locazione 49921. Se lo sprite da cui dovrebbe partire il proiettile è fermo, lo sparo non è attivato.

Il primo carattere dello sprite di sparo è quello immediatamente successivo ai caratteri che definiscono lo sprite da cui parte lo sparo (questo per permettere di modificare automaticamente la forma del proiettile a seconda della direzione da cui viene lanciato). Di questo si deve ovviamente tenere conto quando si creano i caratteri speciali per gli sprite.

Il numero massimo di proiettili che verrà contemporaneamente visualizzato sullo schermo corrisponde al numero di sprites per sparo che sono stati definiti nella matrice A$.

Questi sprites sono i primi della matrice A$. La fine dell'area-sparo è segnalata dall'elemento 2 dell'ultimo sprite per sparo, che deve essere uguale a 255. Questo implica che ci deve essere sempre almeno uno sprite, nell'area sparo della matrice A$, anche quando il programma non prevede la possibilità di sparare. Di questo tiene automaticamente conto l'opzione 1 del programma Basic.

DENUNCIA DEI REDDITI (48K)

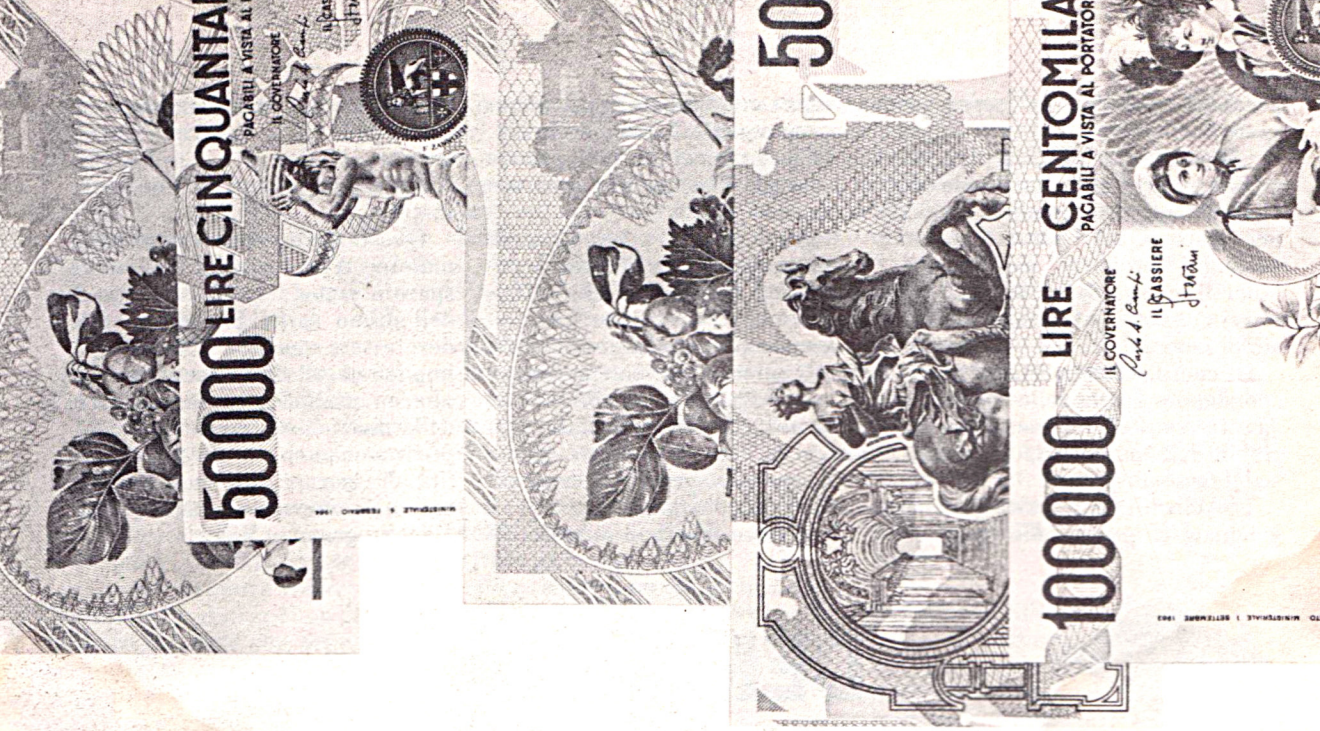
di G. ed F. Previato

Finalmente c'è una soluzione anche per tutti quelli che, una volta all'anno, devono passare una serie di notti in bianco tra conti e tabelle per compilare il famigerato "740"; senza poi contare gli atroci dubbi finali sull'esattezza dei conti, magari dopo aver già consegnato il tutto...

Niente paura, stavolta ci penserà il nostro piccolo (e preciso) Spectrum: il programma non abbisogna di istruzioni particolari: basta inserire in modo corretto tutti i dati richiesti ed il gioco è fatto.

Per stampare i numeri con la giustifica a destra è stato usato un metodo molto semplice: i numeri vengono assegnati in INPUT a delle variabili stringa (che vengono poi convertite in numeri con la funzione VAL), e vengono stampati con una tabulazione uguale a 32 (numero delle colonne video dello Spectrum) meno la lunghezza della stringa-numero. Dalla linea 1260, ad esempio, si trovano le routines per stampare gli oneri deducibili. Per poter esaminare il listato dare come risposta "LNR" alla prima richiesta di INPUT numerico.

```
1260>PRINT AT 11,0;"ILOR e 50% i
mp. arretr.:": INPUT b$: IF b$="
" THEN GO TO 1260
1280 LET Ilor=VAL b$
1300 PRINT AT 11,32-LEN b$;b$
1320 PRINT AT 12,0;"Interessi pa
ssivi        :": INPUT b$: IF b$="
" THEN GO TO 1320
1340 LET Intpass=VAL b$
1360 PRINT AT 12,32-LEN b$;b$
1380 PRINT AT 13,0;"Assicuraz. e
 contr.vol.:": INPUT b$: IF b$="
" THEN GO TO 1380
```



LOAD BANK

Hai programmi originali, esclusivamente pensati e fatti da te? Mandaceli in visione e, se verranno pubblicati, saranno certamente compensati a partire da lire 100 mila in su (a seconda del tipo di programma). Se sei fantasioso, bravo e veloce oltre che ordinato, puoi arrivare a guadagnare un bel gruzzoletto! Naturalmente il tuo nome apparirà stampato sulla rivista come valido collaboratore. Fruga nella tua fantasia e mandaci non solo arcade ma anche utility inedite, giochi di società particolari, interessanti routine in linguaggio macchina ed avventure con grafica (invia sempre anche la soluzione comando per comando in sequenza) allegando anche caratteristiche tecniche dei programmi, listati, stampate di screen etc. Se decidi di inviare un programma (indirizza a Load'n'Run, c.so Vitt. Emanuele 15, 20122 Milano) segui queste regole:

1) salva il programma su entrambi i lati di una cassetta;
2) usa 1 cassetta per ogni programma;
3) scrivi il tuo nome, quello del programma e quello della macchina sulla cassetta;
4) accludi un foglio dattiloscritto dove spiegherai a cosa serve il programma, come si usa, quali tasti usare, lo scopo. Non proteggete i programmi! Dobbiamo guardarli dentro. Non mettete il vostro indirizzo all'interno del programma (basta «by... nome e cognome»).

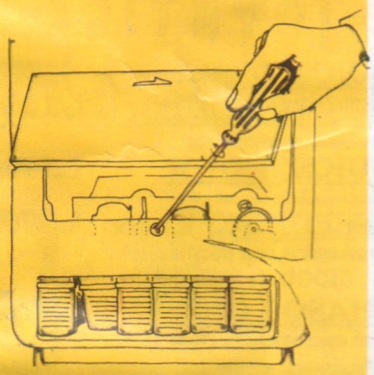
Causa diversità hardware tra le differenti serie di Spectrum, mandaci programmi che non usino il comando IN per la lettura della tastiera; sostituiscilo con un'altra istruzione Basic (per esempio INKEY$) o con una routine in L/M.

Programmi compilati: inviare anche listato sorgente basic. Avventure grafiche: che contengano istruzioni di copy per stampare gli screen!

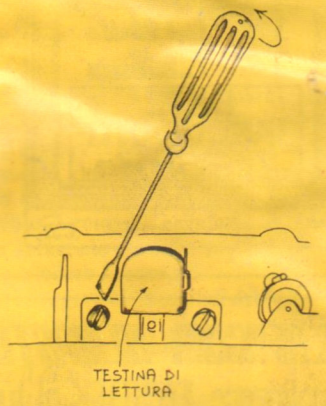
Rispondiamo sempre a tutti e, soprattutto, manteniamo la parola data. Coraggio dunque, fatevi vivi!

ATTENZIONE: I programmi inviati debbono essere assolutamente inediti, a noi ceduti in esclusiva. L'editore si riserva di rifiutare programmi anche già accettati e di non compensare programmi già pubblicati qualora si verificasse che gli stessi programmi (anche quando con titoli diversi) siano stati ceduti o pubblicati da altri.

LE NOSTRE CASSETTE SONO PERFETTE



Se la cassetta non funzionasse al primo colpo, prova a regolare (vedi disegno!) la vite della testina di lettura del registratore fino ad ottenere un segnale perfetto nell'altoparlante!



1) **VOLUME.** Il volume del registratore deve essere sufficientemente alto, in modo da sentire chiaramente dal buzzer del computer il programma in fase di caricamento.
2) **TONO.** Se il registratore ha una sola manopola per la regolazione del tono, essa va spostata tutta verso gli alti (high). Se invece è possibile una equalizzazione, bisogna preferire le frequenze medio-alte.
3) **CANALI.** La cassetta è registrata in mono, cioè su entrambi i canali. In caso di difficoltà è opportuno provare a caricare un solo canale.
4) **AZIMUT.** Per la regolazione dell'azimut, occorre inserire una cassetta nel registratore, premere PLAY e regolare la vite della testina centrale fino ad ottenere dall'altoparlante il massimo segnale.